Main

Table of Contents

Main

Gtk

Doc Writer <christian.popescu@outlook.com> v 1.0, 2019-12-05 :toc: :safe: 0

Gtk

GTK, or the GIMP Toolkit, is a multi-platform toolkit for creating graphical user interfaces.

Links

Gtk home page (https://www.gtk.org/)

Working with Gtk+

MSYS2 / MINGW-64

- Install **GTK3** and its dependencies. Open MSYS2 shell and run:

```
pacman -S mingw-w64-x86_64-gtk3
```

- (recommended): Install GTK core applications
  - **Glade** is a GUI designer for GTK. It lets you design your GUI and export it in XML format. You can then import your GUI from your code using the GtkBuilder API. g++ <programname>.cc -lncurses -I/mingw64/include/ncurses

To install **Glade**:

```
pacman -S mingw-w64-x86_64-glade
```

- **Devhelp** is a help browser. It lets you easily navigate offline in the GTK, glib and gobject API help relative to the version of these libraries installed on your system.

To install **Devhelp**:

```
pacman -S mingw-w64-x86_64-devhelp
```

Links

install on windows (https://www.gtk.org/download/windows.php)

Quick example

C

```
#include <gtk/gtk.h>

static void
print_hello (GtkWidget *widget,
             gpointer   data)
{
  g_print ("Hello World\n");
}

static void
activate (GtkApplication *app,
          gpointer        user_data)
{
  GtkWidget *window;
  GtkWidget *button;
  GtkWidget *button_box;

  window = gtk_application_window_new (app);
  gtk_window_set_title (GTK_WINDOW (window), "Window");
  gtk_window_set_default_size (GTK_WINDOW (window), 400, 400);

  button_box = gtk_button_box_new (GTK_ORIENTATION_HORIZONTAL);
  gtk_container_add (GTK_CONTAINER (window), button_box);

  button = gtk_button_new_with_label ("Hello World");
  g_signal_connect (button, "clicked", G_CALLBACK (print_hello), NULL);
  g_signal_connect_swapped (button, "clicked", G_CALLBACK
(gtk_widget_destroy), window);
  gtk_container_add (GTK_CONTAINER (button_box), button);

  gtk_widget_show_all (window);
}

int
main (int    argc,
      char **argv)
{
  GtkApplication *app;
  int status;

  app = gtk_application_new ("org.gtk.example", G_APPLICATION_FLAGS_NONE);
  g_signal_connect (app, "activate", G_CALLBACK (activate), NULL);
  status = g_application_run (G_APPLICATION (app), argc, argv);
  g_object_unref (app);

  return status;
}
```

Compiling and build:

```
gcc `pkg-config --cflags gtk+-3.0` -o TestGtk TestGtk.c `pkg-config --libs
gtk+-3.0`
```

GTKMM

gtkmm is the official C++ interface for the popular GUI library **GTK+**. Highlights include typesafe callbacks, and a comprehensive set of widgets that are easily extensible via inheritance.

Home Page GTKMM (https://www.gtkmm.org/en/)

Working in MSYS2 / MINGW-64 environment

https://wiki.gnome.org/Projects/gtkmm/MSWindows

- To install the library

  ```
  pacman -S mingw-w64-x86_64-gtkmm3
  ```

- To install

  ```
  pacman -S pkg-config
  ```

Quick example

```
#include <gtkmm.h>

int main(int argc, char** argv)
{
        auto app = Gtk::Application::create(argc, argv);

        Gtk::Window window;
        window.set_default_size(600,400);

        return app->run(window);
}
```

Compile :

```
g++ -std=c++11 TestGtkmm.cc -o TestGtkmm.exe $(pkg-config gtkmm-3.0 --cflags
--libs | sed 's/ -I/ -isystem /g')
```

NCurses

Doc Writer <christian.popescu@outlook.com> v 1.0, 2019-12-02 :toc:

NCurses library

The Ncurses (new curses) library is a free software emulation of curses in System V Release 4.0 (SVr4), and more. It uses **terminfo** format, supports pads and color and multiple highlights and forms characters and function-key mapping.

Links

GNU ncurses (https://www.gnu.org/software/ncurses/)

The ncurses distribution is available at ncurses homepage (https://invisible-island.net/ncurses/)

Working with Ncurses library

MINGW-64

- Set environment variables:

```
export TERMINFO="/usr/share/terminfo"
```

- Simple build command line

```
g++ <programname>.cc -lncurses -I/mingw64/include/ncurses
```

- Simple "Hello world!" program.

```
(Source: ncurses-6.1/doc/html/NCURSES-Programming-HOWTO.html#HELLOWORLD)
```

```
C++
```

```c
#include <ncurses.h>
int main()
{
        initscr(); /* Start curses mode */
        printw("Hello World !!!"); /* Print Hello World*/
        refresh(); /* Print it on to the real screen */
        getch(); /* Wait for user input */
        endwin(); /* End curses mode */
        return 0;
}
```